SOCKETS

Freddy L. Abad Leon

—————————— ◆ ——————————

**COMUNICACION CLIENTE-SERVIDOR**

**SOCKETS**

La comunicación Cliente-Servidor es el modelo descentraliza los procesos y los recursos. Es un sistema donde el cliente es una aplicación, en un equipo, que solicita un determinado servicio y existe un software, en otro equipo, que lo proporciona.

Los servicios pueden ser;

* Ejecución de un programa.
* Acceso a una Base de Datos.
* Acceso a un dispositivo de hardware.

Solo se requiere un medio físico de comunicación entre las máquinas y dependerá de la naturaleza de este medio la vialidad del sistema.

Definición de Socket: designa un concepto abstracto por el cual dos programas (posiblemente situados en computadoras distintas) pueden intercambiarse cualquier flujo de datos, generalmente de manera fiable y ordenada.

Los sockets proporcionan una comunicación de dos vías, punto a punto entre dos procesos. Los sockets son muy versátiles y son un componente básico de comunicación entre interprocesos e intersistemas. Un socket es un punto final de comunicación al cual se puede asociar un nombre.

Para lograr tener un socket es necesario que se cumplan ciertos requisitos:

* Que un programa sea capaz de localizar al otro.
* Que ambos programas sean capaces de intercambiarse información. Por lo que son necesarios tres recursos que originan el concepto de socket:
* Un protocolo de comunicaciones, que permite el intercambio de octetos.
* Una dirección del Protocolo de Red (Dirección IP, si se utiliza el Protocolo TCP/IP), que identifica una computadora.
* Un número de puerto, que identifica a un programa dentro de una computadora.

Con un socket se logra implementar una arquitectura cliente-servidor. la comunicación es iniciada por uno de los programas (cliente). Mientras el segundo programa espera a que el otro inicie la comunicación (servidor). Un Socket es un archivo existente en el cliente y en el servidor.

Si un socket es un punto final de un puente de comunicaron de dos vías entre dos programas que se comunican a través de la red. Normalmente, un servidor funciona en una computadora específica usando un socket con un número de puerto especifico. El cliente conoce el nombre de la maquina (hostname) o el IP, en la cual el servidor está funcionando y el número del puerto con el servidor está conectado.

Si el cliente lanza una demanda de conexión y el servidor acepta la conexión, este abre un socket en un puerto diferente, para que pueda continuar escuchando en el puerto original nuevas peticiones de conexión, mientras que atiende a las peticiones del cliente conectado. El cliente y el servidor pueden ahora comunicarse escribiendo o leyendo en sus respectivos sockets.

Los tipos de socket definen las propiedades de comunicación visibles para la aplicación. Los procesos se comunican solamente entre los sockets del mismo tipo. Existen cinco tipos de sockets.

El Cliente actúa de la siguiente forma:

1. Establece una conexión con el servidor (Crea un socket con el servidor).
2. Mandar mensajes al servidor o Esperar un mensaje de él. (Consultas)
3. Esperar su respuesta o contestarle (existen casos en que este paso no es necesario).
4. Repetir los pasos 2 y 3 mientras sea necesario.
5. Cerrar la conexión con el servidor.

El servidor actúa así:

1. Inicializa un puerto de comunicación, en espera de clientes que intenten conectarse a él (Crea un serverSocket).
2. Una vez que se conecta alguien, crea un hilo de ejecución para este usuario mientras que el thread principal vuelve al paso 1. Esto comúnmente se hace para que el servidor puede atender a varios clientes al mismo tiempo.
3. Se comunica con el cliente mediante el socket creado entre el cliente y él.
4. Espera que el cliente se vaya o lo bota el mismo servidor (Cierra el socket entre ellos) y elimina el thread de comunicación entre ellos.

Las propiedades de un socket dependen de las características del protocolo en el que se implementan. El protocolo más utilizado es TCP, aunque también es posible utilizar UDP o IPX. Gracias al protocolo TCP, los sockets tienen las siguientes propiedades:

1. Orientado a conexión. Se garantiza la transmisión de todos los octetos sin errores ni omisiones.
2. Se garantiza que todo octeto llegará a su destino en el mismo orden en que se ha transmitido.

Los tipos de socket definen las propiedades de comunicación visibles para la aplicación. Los procesos se comunican solamente entre los sockets del mismo tipo.

Existen tres tipos básicos de sockets.

*Socket de flujo*: da un flujo de datos de dos vías, confiable, y sin duplicados sin límites de grabación. El flujo opera en forma parecida a una conversación telefónica. El tipo del socket es SOCK\_STREAM, el cual en el dominio de Internet usa TCP (Transmission Control Protocol).

Socket de datagrama

*Soporta un flujo de mensajes de dos vías*: En un socket de datagrama podría recibir mensajes en diferente orden de la secuencia de la cual los mensajes fueron enviados. Los límites de grabación en los datos son preservados. Los sockets de datagrama operan parecidos a pasar cartas hacia adelante y hacia atrás en el correo. El tipo de socket es SOCK\_DGRAM, el cual en el dominio de internet usa UDP (User Datagram Protocol).

*Socket de paquete secuencial:* Da una conexión de dos vías, secuencial y confiable para datagramas de una longitud fija máxima. El tipo de socket es SOCK\_SEQPACKET. No hay protocolo en Internet implementado para este tipo de socket.

**BIBLIOGRAFIA**

1. Anónimo, “Comunicación Cliente Servidor Sockets”, <http://cesistemas2.blogspot.com/2012/05/211-comunicacion-cliente-servidor.html>